

# Efficient Distributed Non-Asymptotic Confidence Regions Computation over Wireless Sensor Networks

Vincenzo Zambianchi, *Student Member, IEEE*, Michel Kieffer, *Senior Member, IEEE*, Gianni Pasolini, *Senior Member, IEEE*, Francesca Bassi, *Member, IEEE*, and Davide Dardari, *Senior Member, IEEE*

## Abstract

This paper considers the distributed computation of confidence regions tethered to multidimensional parameter estimation under linear measurement models. In particular, the considered confidence regions are non-asymptotic, this meaning that the number of required measurements is finite. Distributed solutions for the computation of non-asymptotic confidence regions are proposed, suited to wireless sensor networks scenarios. Their performances are compared in terms of required traffic load, both analytically and numerically. The evidence emerging from the conducted investigations is that the best solution for information exchange depends on whether the network topology is structured or unstructured. The effect on the computation of confidence regions of information diffusion truncation is also examined. In particular, it is proven that consistent confidence regions can be computed even when an incomplete set of measurements is available.

## I. INTRODUCTION

A wireless sensor network (WSN) consists of energy-limited sensing devices deployed to collaborate in performing a common task. Examples may be the monitoring of an environmental

V. Zambianchi, D. Dardari and G. Pasolini are with the Department of Electrical, Electronics and Information Engineering “G. Marconi” (DEI), University of Bologna, via Venezia 52, Cesena (FC) 47521, Italy. E-mail: {vincenzo.zambianchi, davide.dardari, gianni.pasolini}@unibo.it. M. Kieffer is with CNRS, Supélec, Univ. Paris-Sud, France. E-mail: Michel.Kieffer@lss.supelec.fr. F. Bassi is with ESME Sudria, Supélec, France. E-mail: francesca.bassi@lss.supelec.fr. This work has been supported by the EU funded Newcom# Project.

parameter (*e.g.* temperature or pressure [1]–[3]), the detection of a binary event [4], the estimation of a spatial field [5], the estimation of the coordinates of a signal source [6], etc.

Depending on the specific task requirements (fault tolerance, privacy issues, energy constraints), either a centralized or a distributed approach can be adopted: In the former a central unit is needed, that collects all the information and completes the objective task, whereas in the latter all nodes accomplish the objective task on the basis of the information previously exchanged among them. In the centralized scenario the adoption of efficient routing schemes is of capital importance. Fundamental contributions in this sense are the energy-efficient adaptive clustering proposed in [7] and the routing protocols in [8]–[11], aimed at extending network lifetime.

One of the most studied topic in the WSN literature is the estimation of physical parameters. The literature is mostly focused on the development of some specific estimation techniques, both for the centralized and distributed approaches. Classical maximum likelihood (ML) or least squares (LS) estimators [12] work under the hypothesis of having all the required observations available at one central unit. The scarce robustness to central unit failures and poor network scalability have brought to consideration of distributed approaches. For instance, [13], [14] address recursive weighted LS estimation, alongside a consensus-based algorithm that allows to incorporate information from neighbor nodes in the local estimate. A similar approach is taken within the Bayesian framework in [15]–[17], where consensus-based distributed Kalman filtering is proposed.

The distributed computation of confidence regions has been less considered: In some applications, however, (*e.g.*, in source localization) the derivation of the confidence region is as important as the determination of the estimate. Classical Cramer-Rao-like bounds have been proposed to this purpose in [18]–[21]. Confidence regions can also be derived as a by-product of the application of Kalman filtering [16], [17]. However, strong assumptions on measurement noise (typically Gaussian) are necessary and a good characterization of confidence regions is only possible for a large number of measurements (asymptotic regime).

If we restrict the attention to the centralized setup, the derivation of confidence regions in the non-asymptotic regime has been proved to be possible using, for example, the results in [22]–[25]. Specifically, the methods proposed in [22], [23] allow the central unit to derive a confidence region and a lower bound on the probability that the true value of the estimated

parameter falls within it, whereas the *exact* probability can be obtained using the sign perturbed sums (SPS) algorithm in [24]. In [25], an efficient centralized computation of confidence regions is obtained using interval analysis techniques. Differently from Cramer-Rao-like bounds, these methods do not require precise statistical knowledge of the noise, and work under very mild assumptions on its distribution.

### A. Main Contributions

Some preliminary results on the derivation of exact non-asymptotic confidence regions, in a distributed scenario, appeared in [26]. To ensure that the confidence region computed by each node is similar in shape to the one that would be evaluated in a centralized setup, nodes have to share their local information with one another. The way of diffusing information drastically impacts on the amount of data exchanged. For this reason, several information diffusion strategies are analyzed and compared in the following. A novel information diffusion strategy, named tagged and aggregated sums (TAS), is presented. It exploits the peculiarities of the SPS algorithm, allowing a reduction of the amount of information to be exchanged among nodes. Its performance is compared to that of established information diffusion strategies, such as flooding [1], [27] and consensus algorithms [15], in terms of generated traffic load as well as confidence region volume/traffic trade-off. Performance predictions and simulation results are provided for various topologies. The introduction of the TAS algorithm is one of the novelties of this work.

Constraints on traffic load may lead to information diffusion truncation: Certain nodes might hence compute a confidence region with partial data. However, we prove that consistent non-asymptotic confidence regions can be computed, even starting from an incomplete set of measurements. This constitutes a second theoretical novel contribution to be found in this paper.

The remainder is organized as follows. Section II formulates the confidence region computation problem and recalls the SPS algorithm. Section III presents information diffusion strategies. The computation of non-asymptotic confidence regions, from an incomplete set of measurements, is analyzed in Section IV. Information diffusion techniques are compared on various network topologies in Sections V and VI. Conclusions are drawn in Section VII.

## B. Notation

In this paper, random variables (RVs) are indicated with capital roman or greek letters. Their realizations are denoted by the corresponding lowercase letters. Vectors are denoted by bold letters, being lowercase or uppercase according to their random or deterministic nature, while matrices are indicated with bold capital letters.

## II. PROBLEM FORMULATION

This section recalls the centralized SPS algorithm [24] for the computation of non-asymptotic confidence regions. Consider some spatial field described by the parametric model

$$y_m(\mathbf{x}, \mathbf{p}) = \boldsymbol{\varphi}^T(\mathbf{x}) \mathbf{p}, \quad (1)$$

where  $\mathbf{x} \in \mathbb{R}^{n_x}$  represents some vector of experimental conditions (time, location, ...) under which the field is observed,  $\boldsymbol{\varphi}(\mathbf{x})$  is some regressor function, and  $\mathbf{p}$  is the vector of unknown parameters, belonging to the parameter space  $\mathbb{P} \subset \mathbb{R}^{n_p}$ . For further discussions on the adopted linear model, one may refer to [28] and references therein.

Measurements are taken by a network of  $N$  sensor nodes, spread at random locations  $\mathbf{x}_i \in \mathbb{R}^{n_x}$ ,  $i = 1, \dots, N$ . Each sensor collects its scalar measurement  $y_i$  according to the local measurement model

$$Y_i = y_m(\mathbf{x}_i, \mathring{\mathbf{p}}) + W_i = \boldsymbol{\varphi}_i^T \mathring{\mathbf{p}} + W_i, \quad (2)$$

where  $\boldsymbol{\varphi}_i^T = \boldsymbol{\varphi}^T(\mathbf{x}_i)$  is the regressor vector at  $\mathbf{x}_i$ , assumed to be known,  $\mathring{\mathbf{p}}$  is the true value of the parameter vector and  $W_i$  is a random variable representing the measurement noise. The only assumption on  $W_i$ s is that they are independent from node to node with a distribution, whichever its shape, symmetric with respect to zero.

The aim of this paper is to investigate the distributed derivation of exact non-asymptotic confidence regions, keeping as low as possible the amount of data that has to be exchanged among sensors. As starting point, we recall the centralized SPS algorithm [24] that assumes all measurements and regressors to be known to a central processing unit and returns the exact confidence region around the least squares estimate  $\hat{\mathbf{p}}$  of  $\mathring{\mathbf{p}}$ , obtained as the solution of the normal equations  $\sum_{i=1}^N \boldsymbol{\varphi}_i (y_i - \boldsymbol{\varphi}_i^T \mathbf{p}) = \mathbf{0}$ . Specifically, [24] introduces the unperturbed sum

$$\mathbf{S}_0(\mathbf{p}) = \sum_{i=1}^N \boldsymbol{\varphi}_i (Y_i - \boldsymbol{\varphi}_i^T \mathbf{p}) \quad (3)$$

and the  $m - 1$  sign-perturbed sums, for some  $m$ , with  $2 \leq m \leq N$ ,

$$\mathbf{S}_j(\mathbf{p}) = \sum_{i=1}^N A_{j,i} \varphi_i (Y_i - \varphi_i^T \mathbf{p}), \quad j = 1, \dots, m - 1 \quad (4)$$

where  $A_{j,i} \in \{\pm 1\}$  are independent random signs<sup>1</sup>. Introducing

$$Z_j(\mathbf{p}) = \|\mathbf{S}_j(\mathbf{p})\|_2^2, \quad j = 0, \dots, m - 1, \quad (5)$$

one may define the set

$$\begin{aligned} \Sigma_q &= \{\mathbf{p} \in \mathbb{P} | Z_0(\mathbf{p}) \text{ is not among the } q \text{ largest } Z_j(\mathbf{p})\} \\ &= \left\{ \mathbf{p} \in \mathbb{P} \left| \sum_{j=1}^{m-1} \mathbb{I}(Z_j(\mathbf{p}) - Z_0(\mathbf{p})) \geq q \right. \right\}, \end{aligned} \quad (6)$$

where  $\mathbb{I}(\cdot)$  is the indicator function on positive reals. In [24], it was proven that

$$\text{Prob}(\mathring{\mathbf{p}} \in \Sigma_q) = 1 - \frac{q}{m}. \quad (7)$$

As a consequence  $\Sigma_q$  is a non-asymptotic confidence region with confidence level  $1 - q/m$ .

In the following, the distributed computation of  $\Sigma_q$  will be addressed considering different information diffusion strategies.

### III. INFORMATION DIFFUSION ALGORITHMS

This section describes concurrent procedures for information diffusion adapted to SPS. The purpose is to let each node capable of collecting the largest amount of measurements  $y_i$  and regressors  $\varphi_i$  possibly with the lowest amount of data exchanged in the network. For each presented algorithm, the evolution of the amount of information available at a node  $k$  is described by a table  $\mathbf{R}^{(k)}$ . The construction of  $\mathbf{R}^{(k)}$  and the transmission of information depend on the considered procedure.

#### A. Plain Flooding (PF) Algorithm

When adopting this simple information diffusion strategy [1], [26], [27] each node broadcasts in turn its own measurement and regressor, *i.e.*  $\mathbf{D}^{(k)} = [\varphi_k^T, y_k]$ , as well as those received from other nodes in previous rounds. This strategy is the most trivial one but does not result to be particularly efficient. On lossless networks, it is outperformed by the following one, and is therefore no more considered in the remainder of this work.

<sup>1</sup>A random sign is a symmetric  $\pm 1$  valued random variable taking both values with the same probability  $1/2$ .

$$\mathbf{R}^{(1)} =$$

$\mathbf{D}_1^{(1)}$	1	0	0	0	0	0	0
$\mathbf{D}_2^{(1)}$	0	0	1	0	0	0	0
$\mathbf{D}_3^{(1)}$	0	0	0	0	0	1	0
$\mathbf{D}_4^{(1)}$	0	1	0	0	0	0	0
$\mathbf{D}_5^{(1)}$	0	0	0	1	0	0	0
$\mathbf{D}_6^{(1)}$	0	0	0	0	0	0	1
$\mathbf{D}_7^{(1)}$	0	0	0	0	1	0	0

TABLE I: Table  $\mathbf{R}^{(1)}$  of available information at node  $k = 1$  when MF is used for information diffusion.

### B. Modified Flooding (MF) Algorithm

The main difference between the MF and the PF is that in the former, an information already transmitted by a node is never transmitted again by the same node. This kind of behavior is certainly efficient in terms of amount of data to be transmitted on lossless links.

The MF algorithm generates at runtime a table of contents available at nodes. An example for this is depicted in Table I. This table gathers the information collected at node  $k = 1$  in a network composed of 7 nodes. Each row  $r$  in  $\mathbf{R}^{(k)}$  contains an available information  $\mathbf{D}_r^{(k)}$  and its related tag, indicating the originating node. When performing the MF algorithm, the generic node  $k$  initially fills the first line of  $\mathbf{R}^{(k)}$  with its own local information, *i.e.*,  $\mathbf{D}_1^{(k)} = [\varphi_k^T, y_k]$  and the corresponding tag<sup>2</sup>  $\mathbf{t}_1^{(k)}$  having a single 1 at the  $k$ -th entry. It then broadcasts  $\{\mathbf{D}_1^{(k)}, \mathbf{t}_1^{(k)}\}$  and marks the line as already transmitted. As next step, it collects the data coming from neighbors and inserts in the table this new acquired information, thus creating a set of rows corresponding to its set of neighbor nodes, here denoted as  $\mathcal{N}_k$ . Then it forwards a new data packet containing the data of all lines in  $\mathbf{R}^{(k)}$  which were not marked as already transmitted. This means that the second message that node  $k$  transmits contains  $\{\mathbf{D}_j^{(k)}, \mathbf{t}_j^{(k)}\}_{j \in \mathcal{N}_k}$ . All rows whose data have been transmitted are then marked. The iteration of the procedure yields, at the next transmission step, a message to be transmitted containing only information never previously transmitted. This process terminates when each node in the network has collected the information from all nodes. Section IV analyses the case when all data cannot be gathered at all nodes due, e.g., to

<sup>2</sup>We denote the tag matrix by  $\mathbf{T}^{(k)}$  and its  $r$ -th row by  $\mathbf{t}_r^{(k)}$ .

delay/traffic constraints.

Afterwards, each node is able to compute the perturbed and unperturbed sums in (3) and (4) for any  $\mathbf{p}$ , and hence derive the confidence region. During the first iteration, each node has to transmit a packet containing

$$d_{MF} = n_p + 1 \quad (8)$$

real values. The dimension of successive data packets is an integer multiple of this value, possibly zero.

*Remark 1:* If all nodes agree on their random generators seed, the computed confidence regions are the same at all nodes without any need for transmission of  $A_{j,i}$ . In case this agreement is lacking, still transmission of  $A_{j,i}$  can be avoided, but the shape of confidence regions computed at different nodes may differ.

### C. Tagged and aggregated sums (TAS) Algorithm

Before coming to the detailed description of the TAS algorithm, a preliminary consideration is needed. Expanding a realization of (3) and (4) one gets,

$$\mathbf{s}_0(\mathbf{p}) = \sum_{k=1}^N \boldsymbol{\varphi}_k y_k - \left( \sum_{k=1}^N \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \right) \mathbf{p} \quad (9)$$

$$\mathbf{s}_j(\mathbf{p}) = \sum_{k=1}^N a_{j,k} \boldsymbol{\varphi}_k y_k - \left( \sum_{k=1}^N a_{j,k} \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \right) \mathbf{p}, \quad j = 1, \dots, m-1 \quad (10)$$

The evaluation of (9) and (10) for any value of  $\mathbf{p} \in \mathbb{P}$  does not necessarily require the availability of each individual term but rather of

$$\left\{ \sum_{k=1}^N \boldsymbol{\varphi}_k y_k, \sum_{k=1}^N \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T, \left\{ \sum_{k=1}^N a_{j,k} \boldsymbol{\varphi}_k y_k \right\}_{j=1, \dots, m-1}, \left\{ \sum_{k=1}^N a_{j,k} \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \right\}_{j=1, \dots, m-1} \right\}. \quad (11)$$

Therefore, at each information diffusion step, the available information can be composed into an aggregated sum, reducing the traffic load. This is the peculiarity of the SPS algorithm that can be exploited by both the TAS and the consensus algorithms. The main difficulty lies in avoiding the same term to appear more than once in each sum, independently of network topology. This consideration led to the formulation of the TAS algorithm whose details follow.

The TAS algorithm consists of six phases, namely, i) initialization, ii) reception, iii) distillation, iv) aggregation, v) transmission, and vi) wrap-up, introduced hereafter.

i) *Initialization phase*. During the initialization phase each node  $k \in \{1, \dots, N\}$  creates and transmits a data packet which consists of the first row of its table  $\mathbf{R}^{(k)}$ . This first row is composed of:

- a *data set*  $\mathbf{D}_1^{(k)} = \left\{ \varphi_k y_k, \{\varphi_k \varphi_k^T\}, \{a_{j,k} \varphi_k y_k\}_{j=1, \dots, m-1}, \{a_{j,k} \varphi_k \varphi_k^T\}_{j=1, \dots, m-1} \right\}$ , corresponding to the local quantities related to node  $k$ . This set consists of

$$d_{\text{TAS}} = m \left( n_p + n_p \frac{n_p + 1}{2} \right) \quad (12)$$

real values. This computed dimension takes into account the symmetry of  $\varphi_k \varphi_k^T$ . The dimension of data sets obtained as sums of initial data sets does not vary and stays equal to  $d_{\text{TAS}}$ .

- a *tag vector*  $\mathbf{t}_1^{(k)}$ , that is an all-zero vector except for the  $k$ -th entry where a 1 is located.

After initialization, the reception, distillation, aggregation, and transmission phases are sequentially repeated until a termination condition is met. Within each cycle, new rows  $\{\mathbf{D}_r^{(k)}, \mathbf{t}_r^{(k)}\}$  are possibly added to  $\mathbf{R}^{(k)}$ , with  $r > 1$  representing the row number.

For  $r > 1$ , the  $r$ -th *data set*  $\mathbf{D}_r^{(k)}$  can either contain the local quantities related to another node or the sum of quantities related to several nodes, as specified in  $\mathbf{t}_r^{(k)}$ .

ii) *Reception phase*. During this phase each node collects *messages* transmitted by its neighbors. The message  $\mathbf{m}^{(n)}$  coming from node  $n$ , with  $n \neq k$ , consists of a *data set* and a *tag vector*.

iii) *Distillation phase*. At the end of the reception phase, at each node, the received *tag vectors* and those already stored in  $\mathbf{R}^{(k)}$  are compared to detect whether the received data contains new information. More precisely, the received *tag vector* of each incoming *message* is compared to all the already available *tag vectors*  $\mathbf{t}_r^{(k)}$  contained in  $\mathbf{R}^{(k)}$ . If a received *message* does not contain any new contribution it is discarded, otherwise a new row is added to  $\mathbf{R}^{(k)}$ , containing the new information contribution, that is, the received *message* (*data set+tag vector*) duly polished of already available information.

*Example 1*: if node  $k$  receives a *message* containing the sum of quantities originating from nodes 1, 2, 7, 8, 11 and if it has already rows in  $\mathbf{R}^{(k)}$  containing the information relative to node 1 and to the sum of local quantities of nodes 2 and 7, it can successfully detect the sum of quantities related to nodes 8 and 11 and insert them in the table. Only this distilled information,



$\mathbf{D}_1^{(1)}$	1	0	0	0	0	0	0
$\mathbf{D}_2^{(1)}$	0	0	1	0	0	0	0
$\mathbf{D}_3^{(1)}$	0	0	0	0	0	1	0
$\mathbf{D}_4^{(1)}$	0	1	0	0	0	0	1
$\mathbf{D}_5^{(1)}$	0	0	0	1	0	0	0
$\mathbf{D}_6^{(1)}$	0	0	0	0	1	0	1
$\mathbf{D}_7^{(1)}$	0	1	0	0	1	0	0

TABLE II: Table of available information at node  $k = 1$  when information diffusion is done via the TAS algorithm.

composed of a new *data set* and its corresponding *tag vector* (having the 8th and 11th bits set to 1) is added to  $\mathbf{R}^{(k)}$ .

iv) *Aggregation phase*. To form the next packet to transmit, each node aggregates the information contained in  $\mathbf{R}^{(k)}$ , summing the available *data sets* and merging the related *tag vectors*. The merge is done as follows. When the aggregation phase takes place for the first time, each node  $k$  initializes a *temporary data set*  $\mathbf{D}_T^{(k)}$  and the related  $N$  elements *temporary tag vector*  $\mathbf{t}_T^{(k)}$  with the content of the first row of  $\mathbf{R}^{(k)}$  and marks this row as already merged. Then, the node checks whether the next row contains some information that is already accounted for in  $\mathbf{t}_T^{(k)}$ . If not, it marks it as already merged and sums its corresponding *data set* to  $\mathbf{D}_T^{(k)}$  and updates  $\mathbf{t}_T^{(k)}$ . All rows are then progressively examined. Successive aggregation phases initialize  $\mathbf{D}_T^{(k)}$  and  $\mathbf{t}_T^{(k)}$  as the content of the first never merged row, starting the search from the first row.

*Example 2:* Consider  $\mathbf{R}^{(1)}$  reported in Table II. Node  $k = 1$  has to compose the first *message* that it should transmit. It starts from the first row and initializes  $\mathbf{D}_T^{(1)}$  and  $\mathbf{t}_T^{(1)}$  with the content of the first row. It then marks the first row as already merged. The second row is then examined. As it contains only new information, with respect to the content of  $\mathbf{t}_T^{(1)}$ , its corresponding *data set* is added to  $\mathbf{D}_T^{(1)}$  and its *tag vector* is merged with  $\mathbf{t}_T^{(1)}$ , resulting in  $\mathbf{t}_T^{(1)} = (1, 0, 1, 0, 0, 0)$  and  $\mathbf{D}_T^{(1)} = \mathbf{D}_1^{(1)} + \mathbf{D}_2^{(1)}$ . The same happens for the third, fourth and fifth rows, that are then all marked as already merged. The sixth row contains, instead, information relative to node 7: Node 7 is already contributing to the current  $\mathbf{t}_T^{(1)}$ , thus,  $\mathbf{D}_6^{(1)}$  is not added to  $\mathbf{D}_T^{(1)}$ . Afterwards,  $\mathbf{D}_T^{(1)}$  and  $\mathbf{t}_T^{(1)}$  are transmitted as definitive message, when all rows of the table have been traversed.

When the next aggregation phase takes place,  $\mathbf{D}_T^{(1)}$  and  $\mathbf{t}_T^{(1)}$  are initialized as the content of the first row that has never been merged in the previous aggregation phases: In our example, this happens for the sixth row.

v) *Transmission phase*. The message obtained at the end of the aggregation phase is broadcasted to all neighbor nodes.

The information diffusion stops after a fixed number of transmission phases: On random networks the limit can be set equal to the diameter of the network (as would be the case for any flooding approach).

vi) *Wrap-up phase*. Once the information diffusion expires, the objective, for any node  $k$ , is the computation of (11), which is then used to evaluate (5). This means finding a strategy to combine the rows in  $\mathbf{R}^{(k)}$  to obtain the aggregated data in (11). Two cases are possible: Either  $\text{rank}(\mathbf{T}^{(k)}) = N$  and then a perfect reconstruction of (11) is possible, since each appearing term can be individually retrieved, or  $\text{rank}(\mathbf{T}^{(k)}) < N$  and node  $k$  will try to close as much as possible on (11). This can be realized performing a linear combination of the rows of  $\mathbf{R}^{(k)}$ , aiming at maximizing the amount of data taken into account.

Each node  $k$  will evaluate a linearly weighted sum  $\mathbf{D}_F^{(k)} = \sum_r \hat{b}_{k,r} \mathbf{D}_r^{(k)}$ , where  $\hat{\mathbf{b}}_k$  is the solution of the following constrained optimization problem

$$\hat{\mathbf{b}}_k = \arg \max_{\mathbf{b}_k} \mathbf{b}_k \mathbf{T}^{(k)} \mathbf{1}, \quad (13)$$

$$\text{s.t. } 0 \leq \sum_r b_{k,r} t_{r,i}^{(k)} \leq 1, \quad i = 1, \dots, N. \quad (14)$$

Here,  $t_{r,i}^{(k)}$  are the elements of  $\mathbf{T}^{(k)}$ , with  $r$  and  $i$  denoting the row and column indexes. The solution of (13)-(14) is obtained by linear programming.

The term  $c_{k,i} = \sum_r b_{k,r} t_{r,i}^{(k)}$  in (14) represents the weight of the quantities related to node  $i$ . Since local quantities in (4) cannot contribute more than once, to keep independence among all terms intervening in (4), then it must be  $0 \leq c_{k,i} \leq 1$ , that determines the constraints (14).

*Remark 2:* The TAS algorithm takes some inspiration from network coding techniques [29]–[31]. However, the main difference is that each node does not need to decode, by means of Gaussian elimination, all the individual messages transmitted by the other nodes, but rather the decoding of their sum (possibly of an *incomplete* sum) suffices.

The performance of the TAS algorithm will be investigated in Sections V and VI.

#### D. Consensus Algorithm

Given that the SPS algorithm does not require the single terms appearing in (9) and (10) but rather their sum, a possibility to compute (9) and (10), in a distributed way, is to launch an average consensus algorithm [32]–[35], converging to (11), as recently proposed in [26]. For this information diffusion strategy,  $\mathbf{R}^{(k)}$  is always composed of a single row, storing the consensus state vector. Further details can be found in the referenced papers [26], [32]–[35]. Consensus algorithms will be considered in the numerical results section, anyway we will not put more emphasis since they showed a poor performance in terms of generated traffic load, as investigated in [26].

#### IV. ANALYSIS OF INFORMATION DIFFUSION TRUNCATION

In this section, the effect of truncation of information diffusion is discussed. The objective is to prove that consistent non-asymptotic confidence regions can still be computed via SPS, at all nodes, even when the information diffusion process is stopped before each node has gathered all data.

To achieve this objective, the truncated expressions of (3) and (4) are provided first. Then, some other preliminary definitions and recalls are outlined. Last, a theorem closes the section.

Truncating the information diffusion algorithm entails that (3) and (4) are estimated taking into account only the data actually received by each node. Hence, at node  $k$ , the following quantities are evaluated from the available data

$$\tilde{\mathbf{S}}_{k,0}(\mathbf{p}) = \sum_{i=1}^N c_{k,i} \boldsymbol{\varphi}_i (Y_i - \boldsymbol{\varphi}_i^T \mathbf{p}) \quad (15)$$

$$\tilde{\mathbf{S}}_{k,j}(\mathbf{p}) = \sum_{i=1}^N c_{k,i} A_{j,i} \boldsymbol{\varphi}_i (Y_i - \boldsymbol{\varphi}_i^T \mathbf{p}), \quad (16)$$

where  $j = 1, \dots, m-1$ , and  $c_{k,i} \in \{0, 1\}$ . The coefficients  $c_{k,i}$  reckon with the availability or absence of the  $i$ -th measurement, due to truncation, at node  $k$ .<sup>3</sup>

Note that (15) is the set of normal equations that would be obtained in a centralized context, considering a weighted least-squares estimator, with a diagonal weight matrix  $\mathbf{C}_k =$

<sup>3</sup>The  $c_{k,i}$  differ from the weighting coefficients appearing in [24, Section 2.2]. The difference is that, here, they depend on the measurement index  $i$ . This makes the two forms of weighting completely unrelated.

$\text{diag}(c_{k,1}, \dots, c_{k,N})$ . Similarly, (16) is the sign perturbed sum that would be obtained when considering weighted least-squares. It will be shown that the confidence region, obtained considering (15) and (16) in (6), is still a non-asymptotic confidence region. Reaching completion of the information diffusion algorithm entails that the  $c_{k,i}$  are all equal to one, thus ensuring equivalence with the centralized scenario. In case of truncation, instead, the  $c_{k,i}$  fall in the interval  $[0, 1]$ , their values depending on the applied information diffusion procedure: In case that the TAS or a consensus approach are applied they might take any value in  $[0, 1]$ , otherwise, with flooding, only 0 and 1 are possible values.

Taking the squared norms of (15) and (16), respectively named  $\tilde{Z}_0(\mathbf{p})$  and  $\tilde{Z}_j(\mathbf{p})$ , for  $j = 1, \dots, m-1$ , allows to define the confidence region that is obtained at node  $k$  when truncation occurs, that is,

$$\tilde{\Sigma}_{q,k} = \left\{ \mathbf{p} \in \mathbb{P} \left| \sum_{j=1}^{m-1} \mathbb{I}(\tilde{Z}_j(\mathbf{p}) - \tilde{Z}_0(\mathbf{p})) \geq q \right. \right\}. \quad (17)$$

In order to characterize the consistency of  $\tilde{\Sigma}_{q,k}$ , that relies on an incomplete set of measurements, it is necessary to recall some definitions taken from [24].

*Definition 1 (Symmetric Random Variables):* Given a probability space  $(\Omega, \mathcal{F}, \mathbb{P})$ ,  $\Omega$  being the sample space,  $\mathcal{F}$  the  $\sigma$ -algebra of events, and  $\mathbb{P}$  the probability measure, a real (possibly  $\mathbb{R}^d$ -valued) RV  $X$  is said to be symmetric about the origin 0 (possibly origin vector  $\mathbf{0}$ ) if

$$\forall A \in \mathcal{F} : \mathbb{P}(X \in A) = \mathbb{P}(-X \in A). \quad (18)$$

The following property recalls [24, Lemma 2].

*Property 1:* Let  $A, B_1, \dots, B_k$  be independent, identically distributed (i.i.d.) random signs. Then  $A, AB_1, \dots, AB_k$  are also i.i.d. random signs.

*Definition 2 (Uniformly Ordered Variables):* A finite set of real-valued RVs  $Z_0, Z_1, \dots, Z_{m-1}$  is said to be uniformly ordered if for all permutations  $i_0, i_1, \dots, i_{m-1}$  of indexes  $0, 1, \dots, m-1$ , one has

$$\mathbb{P}(Z_{i_0} < Z_{i_1} < \dots < Z_{i_{m-1}}) = \frac{1}{m!}. \quad (19)$$

Definition 2 states that all orderings are equiprobable. A direct consequence is that, for a set of uniformly ordered RVs  $Z_0, Z_1, \dots, Z_{m-1}$ , each variable  $Z_i$  takes any position in the ordering with probability  $1/m$ .

With the purpose to formulate Lemma 1, introduced in the following, another few more considerations are needed. In this regard, let  $h(Z_0, Z_1, \dots, Z_{m-1}) : \mathbb{R}^m \rightarrow \mathbb{N}_0^{m-1}$  be a function of  $m$  real variables, with  $\mathbb{N}_0^{m-1}$  denoting the set of naturals from 0 to  $m-1$ . The function provides a permutation  $i_0, i_1, \dots, i_{m-1}$ , such that  $Z_{i_0} \leq Z_{i_1} \leq \dots \leq Z_{i_{m-1}}$ . In case of ties between input variables, the permutation is uniquely determined by applying the following rule. Suppose that  $n$  variables are tied: Thus  $n!$  orderings are possible. Then  $h(\cdot)$  provides a reordering choosing among the possible  $n!$  with uniform distribution. Having premised this, when  $h(\cdot)$  takes RVs as inputs, it can be considered as a discrete random variable with  $m!$  possible outcomes, *i.e.*, as many as the number of possible permutations of  $m$  integers.

*Lemma 1 (Uniform Ordering Lemma):* Let  $Z_0, Z_1, \dots, Z_{m-1}$  be real-valued, i.i.d. RVs. Then they are uniformly ordered.

*Proof:* Consider  $h(\cdot)$ , as previously defined. Since  $Z_0, Z_1, \dots, Z_{m-1}$  are i.i.d. the distribution of  $h(Z_{i_0}, Z_{i_1}, \dots, Z_{i_{m-1}})$  is the same for all permutations. Permutations are in number of  $m!$ , hence each of the outcome of  $h(\cdot)$  has probability  $1/m!$ , since the mechanism, by which  $h(\cdot)$  is defined, guarantees that all outcomes are equally possible. This is equivalent to saying that the variables are uniformly ordered. ■

Lemma 1 is a generalization of [24, Lemma 4], that does not hold for discrete RVs, to both continuous and discrete RVs. The need for this extension will appear in the proof of Theorem 1.

Now, one can state the following theorem.

*Theorem 1:* Under the assumption of measurement noises being symmetric RVs and independent across nodes, the confidence level with which the true parameter value  $\mathbf{p}$  falls in the region  $\tilde{\Sigma}_{q,k}$ , yielded at node  $k$ , is

$$\text{Prob}(\mathbf{p} \in \tilde{\Sigma}_{q,k}) = 1 - \frac{q}{m}, \quad (20)$$

for every  $k = 1, \dots, N$ .

*Proof:* Following a similar approach as in [24], the evaluation of (15) and (16) for  $\mathbf{p}$  gives

$$\tilde{\mathbf{S}}_{k,0}(\mathbf{p}) = \sum_{i=1}^N c_{k,i} \varphi_i W_i \quad (21)$$

and

$$\tilde{\mathbf{S}}_{k,j}(\mathbf{p}) = \sum_{i=1}^N c_{k,i} A_{j,i} \varphi_i W_i, \quad (22)$$

with  $j = 1, \dots, m-1$ . The truncation results in a rescaling of measurement noise terms  $W_i$ , since it only depends on the communication links effectively traversed during the information diffusion phase. This rescaling preserves independence as well as symmetry of noise distributions. Consider, further, that from (21) and (22), one can derive

$$\tilde{Z}_{k,0}(\mathring{\mathbf{p}}) = \left\| \sum_{i=1}^N c_{k,i} \varphi_i W_i \right\|_2^2, \quad (23)$$

and

$$\tilde{Z}_{k,j}(\mathring{\mathbf{p}}) = \left\| \sum_{i=1}^N c_{k,i} A_{j,i} \varphi_i W_i \right\|_2^2. \quad (24)$$

These last two expressions may be rewritten highlighting the independent random measurement noise terms  $W_1, \dots, W_N$ , *i.e.*,

$$\tilde{Z}_{k,0}(\mathring{\mathbf{p}}) = f(c_{k,1}W_1, \dots, c_{k,N}W_N), \quad (25)$$

$$\tilde{Z}_{k,j}(\mathring{\mathbf{p}}) = f(c_{k,1}A_{j,1}W_1, \dots, c_{k,N}A_{j,N}W_N), \quad (26)$$

As already pointed out, each  $c_{k,i}W_i$  has a symmetric distribution. By applying Lemma 1 from [24] to the variables in the collection  $\{c_{k,i}W_i\}_{i=1}^N$  and introducing the set of random signs  $\{B_i\}_{i=1}^N$  we can write  $c_{k,i}W_i = B_i(B_i c_{k,i}W_i) = B_i V_i$ , where  $B_i$  and  $V_i = B_i c_{k,i}W_i$  are independent  $\forall i$  [24, Lemma 1]. We can compact (25) and (26) in the single expression

$$\tilde{Z}_{k,j}(\mathring{\mathbf{p}}) = f(D_{j,1}V_1, \dots, D_{j,N}V_N) \quad (27)$$

for  $j = 0, \dots, m-1$ , with  $D_{0,i} \triangleq B_i$  and  $D_{j,i} \triangleq A_{j,i}B_i$  for  $j = 1, \dots, m-1$ . The set of RVs  $\{D_{j,i}\}_{i=1, j=0}^{N, m-1}$  is also a collection of i.i.d. random signs, this deriving from Property 1 applied to the i.i.d. random signs  $B_i$  and  $\{A_{j,i}\}_{j=1}^{m-1}$ . Now fix a realization for  $\{V_i\}_{i=1}^N$ , indicated as  $\{v_i\}_{i=1}^N$ . Conditioning on  $\{V_i\}_{i=1}^N = \{v_i\}_{i=1}^N$ ,  $\left\{ \tilde{Z}_{k,j}(\mathring{\mathbf{p}}) \middle| \{V_i\}_{i=1}^N = \{v_i\}_{i=1}^N \right\}_{j=0}^{m-1}$  is a collection of discrete, real-valued, and i.i.d. RVs, since  $\{D_{j,i}\}_{i=1, j=0}^{N, m-1}$  is a collection of i.i.d. random signs. Applying Lemma 1 to  $\left\{ \tilde{Z}_{k,j}(\mathring{\mathbf{p}}) \middle| \{V_i\}_{i=1}^N = \{v_i\}_{i=1}^N \right\}_{j=0}^{m-1}$  leads to the consideration that these variables are uniformly ordered. This implies that the RV  $\tilde{Z}_{k,0}(\mathring{\mathbf{p}}) \middle| \{V_i\}_{i=1}^N = \{v_i\}_{i=1}^N$  takes each position in the ordering with probability  $1/m$ . The conclusion is that it is not among the  $q$  largest  $\tilde{Z}_{k,j}(\mathring{\mathbf{p}}) \middle| \{V_i\}_{i=1}^N = \{v_i\}_{i=1}^N$ ,  $j = 0, \dots, m-1$ , with probability  $1 - q/m$ . Since this probability value is independent of the particular realization of  $\{V_i\}_{i=1}^N$ , one can apply [24, Lemma 3] to

say that  $\text{Prob}(\hat{\mathbf{p}} \in \tilde{\Sigma}_{q,k}) = \text{Prob}\left(\tilde{Z}_{k,0}(\hat{\mathbf{p}}) \text{ is not among the } q \text{ largest } \tilde{Z}_{k,j}(\hat{\mathbf{p}})\right) = 1 - \frac{q}{m}$  is valid also when not conditioning on noise realizations. This concludes the proof. ■

*Remark 3:* We introduced Lemma 1 to prove that i.i.d. discrete RVs are uniformly ordered. From [24, Lemma 4], one can draw this conclusion only for continuous variables. The here presented Lemma 1 generalizes [24, Lemma 4]. This copes with the discrete RVs, that are appearing when noise realizations are fixed, as done in the proof.

*Remark 4:* When at sensor node  $k$  there is only a single non-zero coefficient,  $c_{k,k} = 1$ , meaning that truncation in information diffusion occurred before node  $k$  could gain knowledge about any other sensor than itself, then, the  $m \times N$  matrix formed by all random signs  $A_{j,i}$  participating in the confidence region computation at node  $k$  has only one column filled with values  $\{-1, 1\}$ , while all the remaining ones are filled with zeros. Its rank is hence equal to 1 and the norms  $\tilde{Z}_{k,j}(\mathbf{p})$  are all equal independently of  $j$  and for any value of  $\mathbf{p}$ . This is certainly the case for which the highest number of ties occurs, nevertheless, choosing at random for the reordering, yields a random confidence region, covering a percentage equal to  $1 - q/m$  of the initial search space. The computed confidence region keeps again the same level of confidence  $1 - \frac{q}{m}$ , as stated by Theorem 1. This observation gives an insight on the reason why the shape of confidence regions is affected by information availability.

## V. TRAFFIC LOAD ON VARIOUS NETWORK TOPOLOGIES

For a fair comparison of different information diffusion strategies, the network traffic burden has to be characterized. The algorithms are compared on specific topologies, such as random trees, with binary trees as a special case, and clustered networks, that are the most commonly used in practical applications [3]. In Section VI, completely random networks will also be considered.

Before entering into the details of our analytical investigation, let us recall that  $d_{\text{TAS}}$  and  $d_{\text{MF}}$ , respectively given by (12) and (8), denote the numbers of real-valued scalars that a single data is composed of when the TAS or the MF algorithm are considered.

The remainder of this section is divided into as many subsections as the considered topologies.

### A. Random Tree Topology

Consider a random tree topology, *i.e.*, a tree where each node has a random number of sons, possibly zero. The number of nodes forming the network is considered equal to  $N$ . The levels

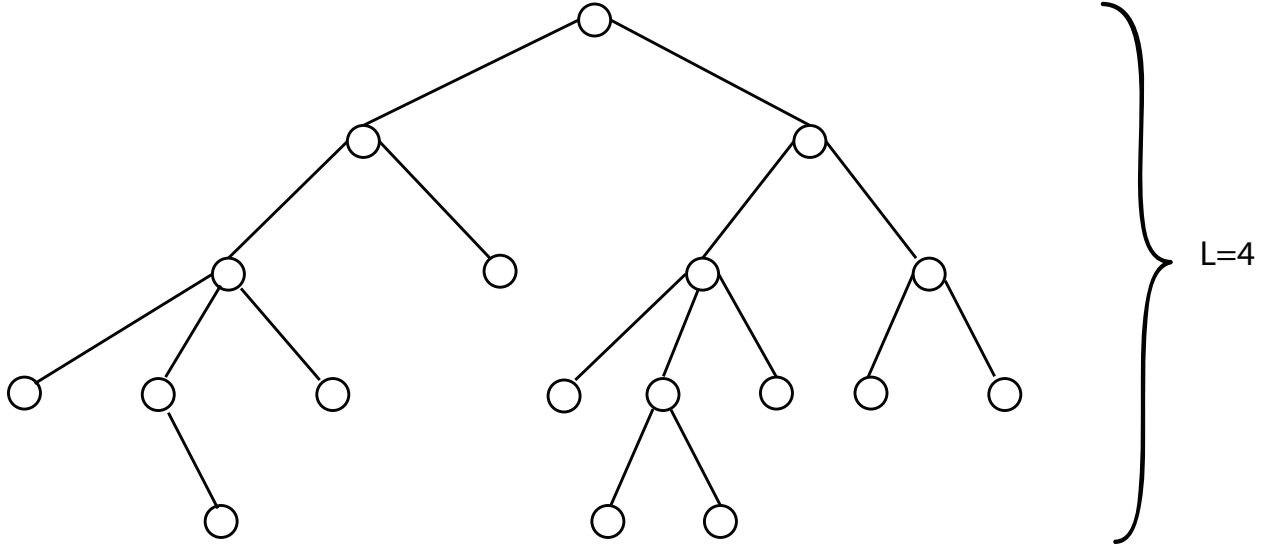


Fig. 1: A random tree topology. Some of the random variables, that describe it, take values  $\Lambda(0) = 1$ ,  $\Lambda(1) = 2$ ,  $\Lambda(2) = 4$ ,  $\bar{\Lambda}(2) = 1$ ,  $\Lambda(3) = 8$ ,  $\bar{\Lambda}(3) = 6$ , etc.

in the tree are indicated by  $\ell$ , with  $\ell \geq 0$ .  $L$  denotes the lowest level in the tree. The set of nodes in the  $\ell$ -th level of the tree is denoted as  $\mathbb{L}_\ell$ , having cardinality  $\Lambda(\ell)$ , which is a RV. Nevertheless,  $\Lambda(0) = 1$  and is not random, since the tree is single rooted. Moreover, the set of nodes, in the  $\ell$ -th level in the tree, that are not parents to any nodes, is denoted by  $\bar{\mathbb{L}}_\ell$  and its cardinality is a RV denoted by  $\bar{\Lambda}(\ell)$ .

1) *TAS algorithm*: The TAS algorithm of Section III-C does not assume any ordering in the network on which it should run. On a random tree, however, it is possible to simplify it making nodes transmit much less frequently than required on an unstructured random topology.

During each transmission phase a single level of the tree is active. Only nodes in this level can transmit. Starting from level  $L$ , each node in  $\mathbb{L}_L$  has to broadcast its own local data. Then, parent nodes distil and aggregate the received quantities with their own ones and broadcast. The process is repeated until the tree root is reached. The tree is then traveled backwards, from Level 0 to Level  $L$ , making the complete sum available to all nodes. This way of operating ensures that an exact retrieval of the entire sum is possible and that no truncation occurs, if the procedure is completed. Nodes participate only in the (at most two) rounds of transmission involving the level they belong to. The number of data that must be transmitted when employing the TAS



algorithm is a discrete RV given by

$$N_{\text{TAS}}^{\text{RT}} = \sum_{\ell=0}^L \Lambda(\ell) d_{\text{TAS}} + \sum_{\ell=1}^{L-1} \Lambda(\ell) d_{\text{TAS}} - \sum_{\ell=1}^{L-1} \bar{\Lambda}(\ell) d_{\text{TAS}}. \quad (28)$$

$N_{\text{TAS}}^{\text{RT}}$  consists of the number of data transmitted when traversing the tree from level  $L$  to the root, included, plus the amount of data required by the backwards travel. The last term in (28) is related to nodes without sons, which do not transmit anything when the travel backwards is performed.

2) *MF algorithm*: For the MF, one instead gets,

$$\begin{aligned} N_{\text{MF}}^{\text{RT}} &= \Lambda(L) d_{\text{MF}} + (\Lambda(L) + \Lambda(L-1)) d_{\text{MF}} + \dots + \sum_{\ell=0}^L \Lambda(\ell) d_{\text{MF}} + N (\Lambda(1) - \bar{\Lambda}(1)) d_{\text{MF}} \\ &\quad - \sum_{\ell=2}^L \Lambda(\ell) d_{\text{MF}} - (\Lambda(1) - \bar{\Lambda}(1)) d_{\text{MF}} + \dots + N (\Lambda(L-1) - \bar{\Lambda}(L-1)) d_{\text{MF}} \\ &\quad - \Lambda(L) d_{\text{MF}} - (\Lambda(L-1) - \bar{\Lambda}(L-1)) d_{\text{MF}} \\ &= \Lambda(L) d_{\text{MF}} + (\Lambda(L) + \Lambda(L-1)) d_{\text{MF}} + \dots + \sum_{\ell=0}^L \Lambda(\ell) d_{\text{MF}} + (N-1) (\Lambda(1) - \bar{\Lambda}(1)) d_{\text{MF}} \\ &\quad - \sum_{\ell=2}^L \Lambda(\ell) d_{\text{MF}} + \dots + (N-1) (\Lambda(L-1) - \bar{\Lambda}(L-1)) d_{\text{MF}} - \Lambda(L) d_{\text{MF}} \\ &= \sum_{\ell=0}^L \Lambda(\ell) d_{\text{MF}} + \Lambda(L) d_{\text{MF}} + N \sum_{\ell=1}^{L-1} (\Lambda(\ell) - \bar{\Lambda}(\ell)) d_{\text{MF}} + \sum_{\ell=1}^{L-1} \bar{\Lambda}(\ell) d_{\text{MF}}, \end{aligned} \quad (29)$$

where  $\Lambda(L) d_{\text{MF}} + (\Lambda(L) + \Lambda(L-1)) d_{\text{MF}} + \dots + \sum_{\ell=0}^L \Lambda(\ell) d_{\text{MF}}$  is the amount of data transmitted in the forward travel,  $N (\Lambda(1) - \bar{\Lambda}(1)) d_{\text{MF}}$  is the amount of data (proportional to  $N$ ) that nodes, with sons, in level 1 would transmit when the tree is traveled backwards, as if no forward travel was ever performed, and  $\sum_{\ell=2}^L \Lambda(\ell) d_{\text{MF}} + (\Lambda(1) - \bar{\Lambda}(1)) d_{\text{MF}}$  is the amount of data that has to be subtracted from the previous one, since these data have already been transmitted in the forward travel. The other terms can be similarly explained.

3) *Comparison*: TAS is more efficient than MF when  $N_{\text{TAS}}^{\text{RT}} < N_{\text{MF}}^{\text{RT}}$ , i.e., when

$$\begin{aligned} \sum_{\ell=0}^L \Lambda(\ell) (d_{\text{TAS}} - d_{\text{MF}}) - \sum_{\ell=1}^{L-1} \bar{\Lambda}(\ell) (d_{\text{TAS}} + d_{\text{MF}}) - \Lambda(L) d_{\text{MF}} \\ - N \sum_{\ell=1}^{L-1} (\Lambda(\ell) - \bar{\Lambda}(\ell)) d_{\text{MF}} + \sum_{\ell=1}^{L-1} \Lambda(\ell) d_{\text{TAS}} < 0 \end{aligned}$$

$$\begin{aligned}
& N(d_{\text{TAS}} - d_{\text{MF}}) - \Lambda(L)d_{\text{MF}} + (N - \Lambda(L) - 1)d_{\text{TAS}} \\
& - N(N - \Lambda(L) - 1)d_{\text{MF}} + \sum_{\ell=1}^{L-1} \bar{\Lambda}(\ell) ((N - 1)d_{\text{MF}} - d_{\text{TAS}}) < 0, \quad (30)
\end{aligned}$$

that is,

$$\begin{aligned}
& (d_{\text{MF}} + d_{\text{TAS}} - Nd_{\text{MF}}) \Lambda(L) > N(d_{\text{TAS}} - d_{\text{MF}}) + (N - 1)d_{\text{TAS}} - N(N - 1)d_{\text{MF}} \\
& + \sum_{\ell=1}^{L-1} \bar{\Lambda}(\ell) ((N - 1)d_{\text{MF}} - d_{\text{TAS}}). \quad (31)
\end{aligned}$$

In case  $N$  is finite, (31) is satisfied with a probability that is not easily evaluated. Nevertheless, an asymptotic consideration can be done. Firstly, when  $N \rightarrow \infty$ , we assume that  $L \rightarrow \infty$ . This is precisely the case when the area on which nodes are deployed is increasing with  $N$ , due to coverage extension purposes, or if the communication range  $d_{\text{comm}}$  is diminishing with  $N$ , due to interference mitigation purposes. Moreover, one has  $\sum_{\ell=1}^{L-1} \bar{\Lambda}(\ell) < N - \Lambda(L) - L$ , since the number of nodes without sons, in all levels except 0 and  $L$ , cannot exceed the total number of nodes deprived of the number of nodes at level  $L$  and of at least one node for each of the  $L$  levels from 0 to  $L - 1$  (if no nodes are present in a level there cannot be any further levels). Then, passing to the limit for  $N \rightarrow +\infty$ , if

$$\begin{aligned}
& \lim_{N \rightarrow \infty} \{N(d_{\text{TAS}} - d_{\text{MF}}) + (N - 1)d_{\text{TAS}} - N(N - 1)d_{\text{MF}} \\
& + ((N - 1)d_{\text{MF}} - d_{\text{TAS}}) (N - \Lambda(L) - L) - (d_{\text{MF}} + d_{\text{TAS}} - Nd_{\text{MF}}) \Lambda(L)\} < 0 \quad (32)
\end{aligned}$$

holds, *i.e.*,

$$\lim_{N \rightarrow \infty} (d_{\text{TAS}} - d_{\text{MF}}) + \frac{(L - 1)}{N} d_{\text{TAS}} - \frac{L(N - 1)}{N} d_{\text{MF}} < 0, \quad (33)$$

then also (31) holds asymptotically. Since  $L$  also goes to  $+\infty$ , (33) is verified and thus (31) holds for all values of  $\Lambda(L)$ , hence with probability 1. Moreover, this is true for all values of the problem dimensions, *i.e.*,  $n_p$  and  $m$ .

## B. Binary Tree Topology

A deterministic *complete binary* tree topology, that is a tree where each node has exactly two sons apart from nodes in level  $L$  that do not have any, is now considered. Assuming that the binary tree consists of  $L$  levels entails  $N = 2^{L+1} - 1$ .

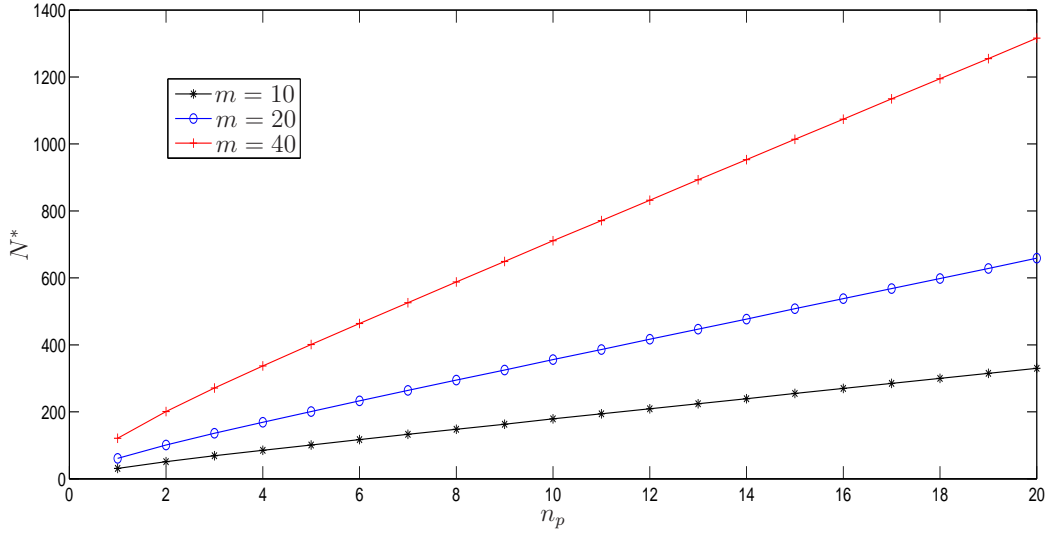


Fig. 2: Critical value  $N^*$ , as a function of  $n_p$ , on binary trees, for several values of  $m$ .

1) *TAS algorithm*: For the TAS algorithm, the total number of required data communications is deduced from (28)

$$\begin{aligned} N_{\text{TAS}}^{\text{BT}} &= (2^L + 2^{L-1} + \dots + 2^1 + 1 + 2^1 + \dots + 2^{L-1}) d_{\text{TAS}} \\ &= \left( 2^{L+1} - 2 + \frac{1}{2} (2^{L+1} - 2) \right) d_{\text{TAS}} = \left( \frac{3N - 3}{2} \right) d_{\text{TAS}}. \end{aligned} \quad (34)$$

2) *MF algorithm*: The total number of data communications required by the MF algorithm is deduced from (29)

$$N_{\text{MF}}^{\text{BT}} = (2^{L+1} - 1) d_{\text{MF}} + 2^L d_{\text{MF}} + (2^{L+1} - 1) \sum_{i=1}^{L-1} 2^i d_{\text{MF}} = \frac{N^2 + 1}{2} d_{\text{MF}}. \quad (35)$$

3) *Comparison*: On a binary tree, TAS is more efficient than MF when

$$\frac{3N - 3}{2} d_{\text{TAS}} < \frac{N^2 + 1}{2} d_{\text{MF}}. \quad (36)$$

Using (12) and (8) one obtains the following condition

$$(N^2 + 1) K_1 - 3N + 3 > 0, \quad (37)$$

where  $K_1 = \frac{n_p + 1}{(n_p + n_p \frac{n_p + 1}{2})m}$ . For sufficiently large  $N$ , (37) is always satisfied, disregarding  $n_p$  and  $m$ . Moreover, and unlike in the random tree case, given  $n_p$  and  $m$ , it is possible to derive

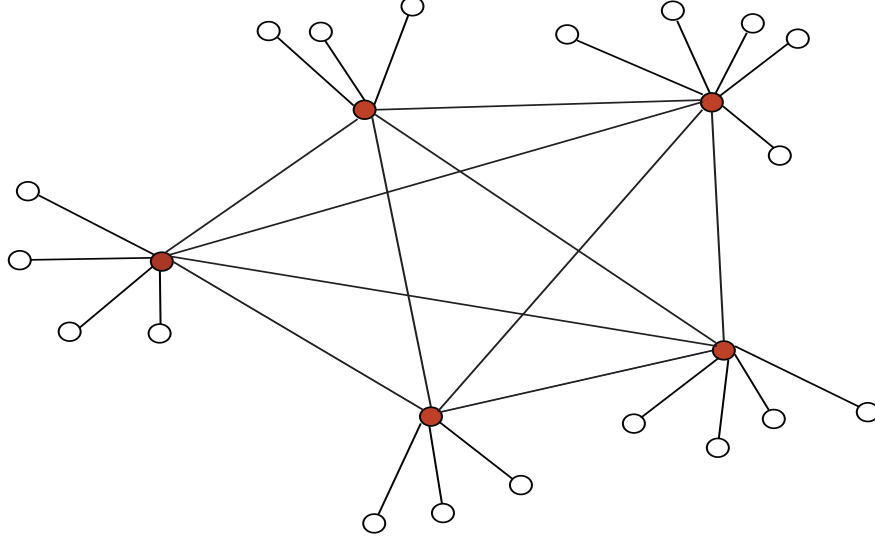


Fig. 3: A clustered topology. Clusterheads are indicated in red.

the value

$$N^* = \frac{3 + \sqrt{9 - 4K_1(3 + K_1)}}{2K_1}, \quad (38)$$

for which TAS is more efficient than MF. Fig. 2 represents  $N^*$  as a function of  $n_p$ , considering  $m = 10, 20, 40$ . The behaviour is not exactly linear, as it can be easily verified by derivation of (38), but rapidly tends to be such: In fact, when  $n_p$  grows large,  $K_1 \approx \frac{2}{n_p m}$  and  $N^* \approx \frac{n_p m}{4} \left[ 3 + \sqrt{9 - \frac{8}{n_p m} \left( 3 + \frac{2}{n_p m} \right)} \right] \approx \frac{3}{2} n_p m$ .

### C. Clustered Topology

Consider a clustered network, formed by  $N$  nodes, structured on a single level of hierarchy (see Fig. 3). The network is hence assumed to be divided in  $n_c$  clusters. The  $i$ -th cluster comprises a random number of nodes  $N_i^c$ , including the clusterhead, that is the special node responsible for aggregating the local data of its sons. The subnetwork formed by clusterheads is considered to be fully connected: Clusterheads can directly communicate to one another. Moreover, each node in a cluster is assumed to directly communicate with its clusterhead (and vice-versa).

1) *TAS algorithm*: On this topology, the TAS algorithm transmission phases can be organized as follows. At the beginning, all nodes, with the exception of clusterheads, transmit their local data to the clusterheads. Then each clusterhead aggregates the local data of all nodes in its

cluster. Successively, clusterheads transmit to all other clusterheads their aggregated data. Since the network of clusterheads is fully connected, a single broadcast transmission for each of the clusterhead suffices for all clusterheads being capable to construct the completely aggregated data. The amount of scalar data, that has to be transmitted, is thus

$$\begin{aligned} N_{\text{TAS}}^{\text{cc}} &= ((N - n_c) + n_c + n_c) d_{\text{TAS}} \\ &= (N + n_c) d_{\text{TAS}}. \end{aligned} \quad (39)$$

This accounts for the initial  $N - n_c$  transmissions and the subsequent actions of clusterheads, that should broadcast to each other the partially aggregated data and then broadcast, towards nodes forming their cluster, the completely aggregated data.

2) *MF algorithm*: All nodes in a cluster can overhear broadcast transmissions operated by the corresponding clusterhead. Therefore, the amount of data to be transmitted when employing the MF algorithm is

$$\begin{aligned} N_{\text{MF}}^{\text{cc}} &= ((N - n_c) + N + (n_c - 1) N) d_{\text{MF}} \\ &= (N - n_c + n_c N) d_{\text{MF}}. \end{aligned} \quad (40)$$

This is because all nodes, apart from clusterheads, initially transmit their local information to clusterheads, giving rise to  $(N - n_c) d_{\text{MF}}$  transmitted scalar data. Then clusterheads broadcast the received data and their own, this forming a total flow of  $N d_{\text{MF}}$  scalar data. At this point, all nodes in each cluster are completely informed about data related to their respective cluster. Finally, there is a backwards transmission during which each clusterhead is transmitting towards its cluster all the  $N d_{\text{MF}}$  scalar data except the ones that it previously transmitted, this being equivalent to further  $(n_c - 1) N d_{\text{MF}}$  transmitted scalars, composed of  $n_c$  clusterheads transmitting not  $N$ , but  $(N - N_i^c) d_{\text{MF}}$  scalar data, *i.e.*, a total of  $\sum_{i=1}^{n_c} (N - N_i^c) d_{\text{MF}} = (n_c N - N) d_{\text{MF}}$ .

3) *Comparison*: TAS is better than MF when  $N_{\text{TAS}}^{\text{cc}} < N_{\text{MF}}^{\text{cc}}$ , *i.e.*, when

$$\begin{aligned} (N - n_c + n_c N) d_{\text{MF}} - (N + n_c) d_{\text{TAS}} &> 0 \\ \left( 1 + \frac{n_c(N - 2)}{N + n_c} \right) \frac{d_{\text{MF}}}{d_{\text{TAS}}} &> 1. \end{aligned} \quad (41)$$

Here  $n_c$  is the degree of freedom, in lieu of  $L$  in the tree topologies. Assuming that, due to coverage extension or interference mitigation purposes,  $n_c$  grows to  $\infty$  with  $N$  going to  $\infty$ , one

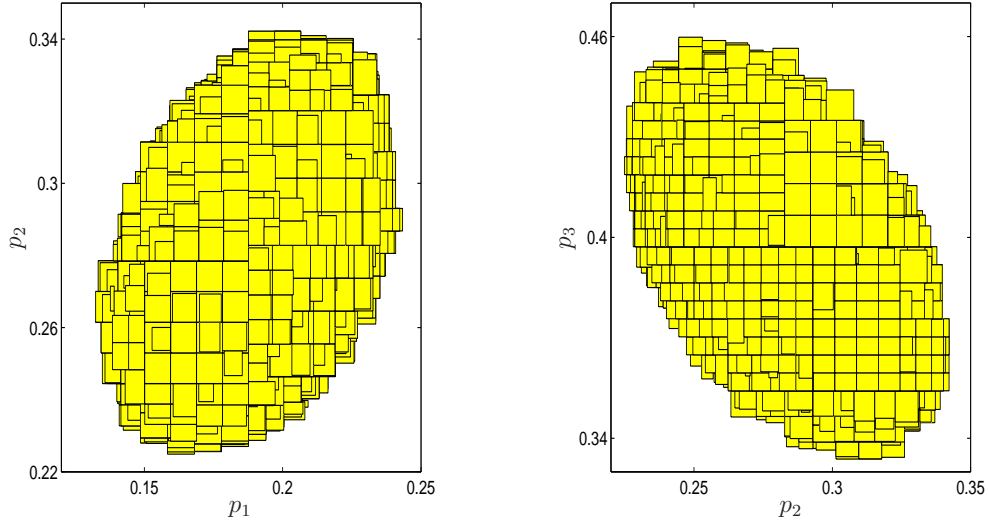


Fig. 4: Projections of the 90% confidence region computed at node 1 after 4 consensus iterations. A random unstructured network of 100 nodes is considered.

has

$$\lim_{N \rightarrow +\infty} \left( 1 + \frac{n_c N - 2n_c}{N + n_c} \right) \frac{d_{\text{MF}}}{d_{\text{TAS}}} = \lim_{N \rightarrow +\infty} n_c \frac{d_{\text{MF}}}{d_{\text{TAS}}} = +\infty, \quad (42)$$

independently on  $n_p$  and  $m$ . Thus, TAS is asymptotically better than MF.

## VI. SIMULATION RESULTS

In this section, all simulations results have been obtained assuming lossless links while confidence regions are evaluated with the interval analysis techniques described in [25].<sup>4</sup> The Intlab package [36] is employed for intervals computations.

We start with a numerical investigation of the effect of truncation in information diffusion on the shape of the confidence region. To this purpose, we instantiate a random unstructured network of  $N = 100$  nodes, uniformly distributed over a unit area, and consider a true parameter value  $\mathring{\mathbf{p}} = [p_1, p_2, p_3] = [0.2, 0.3, 0.4]$ . The inter node communication range is set to  $d_{\text{comm}} =$

<sup>4</sup>These techniques allow for the computation of tight outer approximations of confidence regions via contraction of the initial search space. The contraction halting criterion may be set such that single box outer approximations are obtained, instead of multiple boxes outer approximations. For the sake of simplicity and with abuse of terminology, in the remainder, ‘confidence regions’ is used in lieu of ‘outer approximations of confidence regions’, if not otherwise specified.

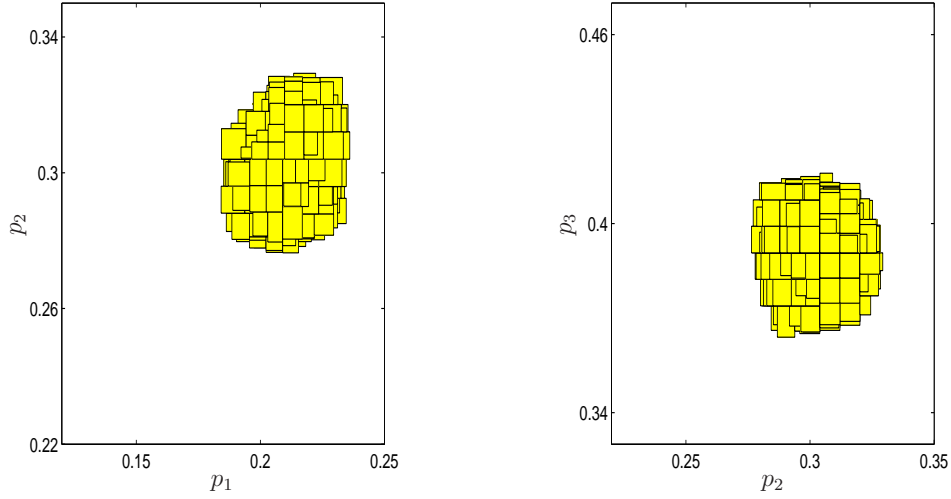


Fig. 5: Projections of the 90% confidence region computed at node 1 after 30 consensus iterations. A random unstructured network of 100 nodes is considered.

$\sqrt{\frac{\log_2 N}{2N}}$ . According to [37], this range guarantees almost sure connectivity of a network of  $N$  nodes, deployed on a finite area. A truncated Metropolis consensus algorithm [26], [32], [34] is considered for the distributed computation of confidence regions. Similar results may be obtained also for the other information diffusion strategies, considered in Section III. Figs. 4 and 5 show the confidence region computed at node 1 after 4 and 30 iterations, respectively. The reduction in terms of volume is quite evident in the second case, while we underline that the confidence level is the same.

In order to compare the TAS and the MF algorithms, we consider random trees, clustered networks, and random unstructured topologies, for the same order of magnitude in terms of number of nodes.

For what concerns the analysis on random trees, we build a spanning tree on top of a random unstructured network, setting  $d_{\text{comm}}$  as earlier done. For each  $N$  (see the horizontal axis in Fig. 6), 100 connected network realizations are instantiated. TAS and MF are compared in terms of the required number of data to be transmitted. The success rate of TAS is the percentage of network realizations that proved favorable to TAS and it is shown in Fig. 6 as a function of  $N$ , for several  $n_p$  values. It can be noticed that, as foreseen in the theoretical analysis in Section V,

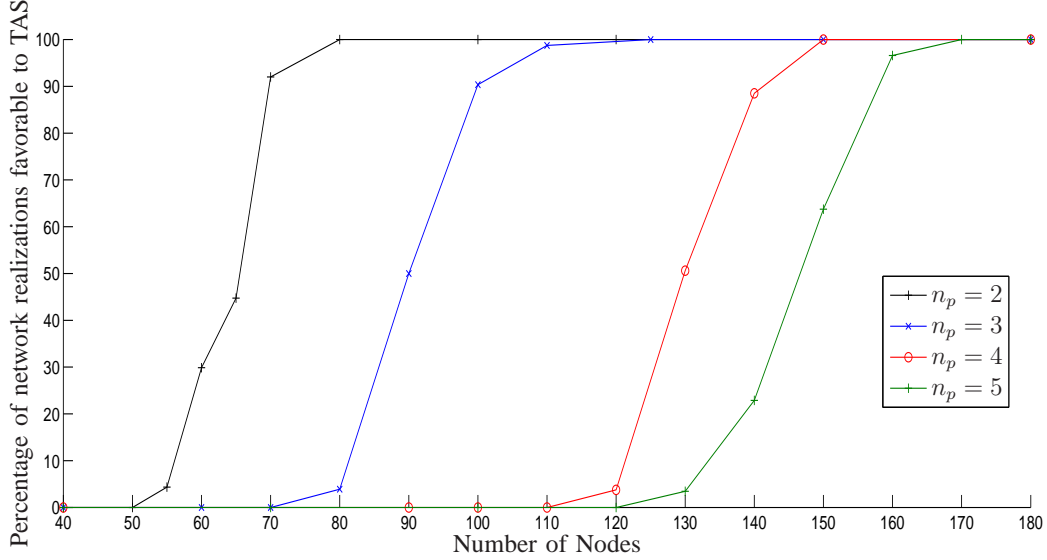


Fig. 6: Percentage of network realizations favorable to TAS, in terms of required data exchanges, compared to MF, as a function of the number of nodes forming a random tree topology for different values of  $n_p$ . 100 random tree realizations are considered for each value of  $N$ .

there always exists a threshold value of  $N$ , depending on  $n_p$ , above which the TAS outperforms the MF algorithm, *i.e.*, the percentage closes to 100.

We now investigate the trade-off between the confidence region volume and the amount of per node transmitted data. Fig.7 shows the average volume of the 90% confidence region as a function of the average amount of data that need to be communicated by each node. The volume and data amount are averaged across all nodes and across 100 random tree realizations, while simulation parameters are set to  $n_p = 2$ ,  $q = 1$ ,  $N = 100$  and  $m = 10$ . Fig. 7 allows to know which is the amount of data that need to be transmitted by each node on average to obtain a given confidence region average volume. Each pair of coordinates corresponds to one transmission round. It is easily seen that the TAS algorithm outperforms the MF for meaningful small volume values, in terms of the amount of per node transmitted data.

Similar results can be obtained on clustered networks. The number of clusters is set to  $n_c = 20$  and the average number of per cluster nodes is set to  $\mathbb{E}[N_i^c] = 7$ . The parameter dimension is  $n_p = 2$ , while  $q = 1$  and  $m = 10$ . In particular, Fig. 8 shows the average volume, across nodes and 100 clustered network realizations, of the confidence region. Here the number of computed



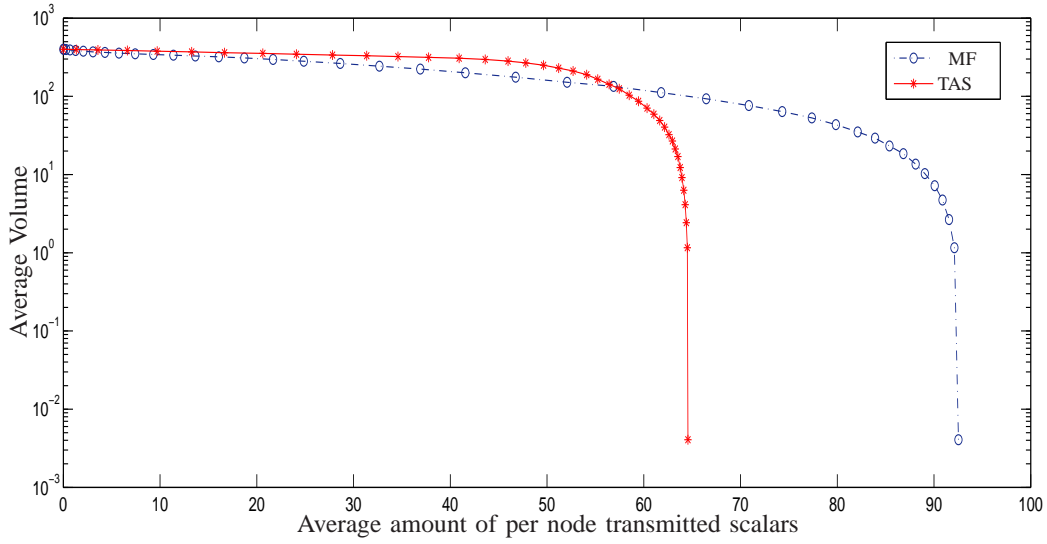


Fig. 7: Average volume, across nodes and 100 random tree realizations, of the 90% confidence region. Simulation parameters are set to  $N = 100$ ,  $n_p = 2$ ,  $q = 1$ , and  $m = 10$ .

pairs volume-amount of data is much lower than for random trees, due to the fewer transmission rounds. The average amount of per node transmitted data, needed to obtain meaningful small volumes, is lower when employing the TAS algorithm, as it was on random trees.

Finally, consider a random unstructured network, setting  $N = 100$  and  $n_p = 3$ . As shown in Fig. 9, in this case it is the MF algorithm that behaves better than TAS, providing lower volume values for the same amount of data. For comparison, it is also shown how both the MF and the TAS algorithm outperform the state of the art consensus algorithms, independently of the considered consensus matrix (Metropolis [32] or Perron [15]).

This section confirms the general behavior that was highlighted in Section V: On structured topologies, such as random trees and clustered networks, there is an advantage in employing the TAS algorithm when the network dimension is sufficiently large, and this independently of  $n_p$ . On unstructured networks of comparable size, the MF produces the best results, but, in any case, the absolute amount of per node transmitted data is much larger than in structured networks. This suggests the adoption of structured networks, together with the TAS algorithm for the distributed computation of confidence regions, when the network traffic load for data diffusion is particularly critical.

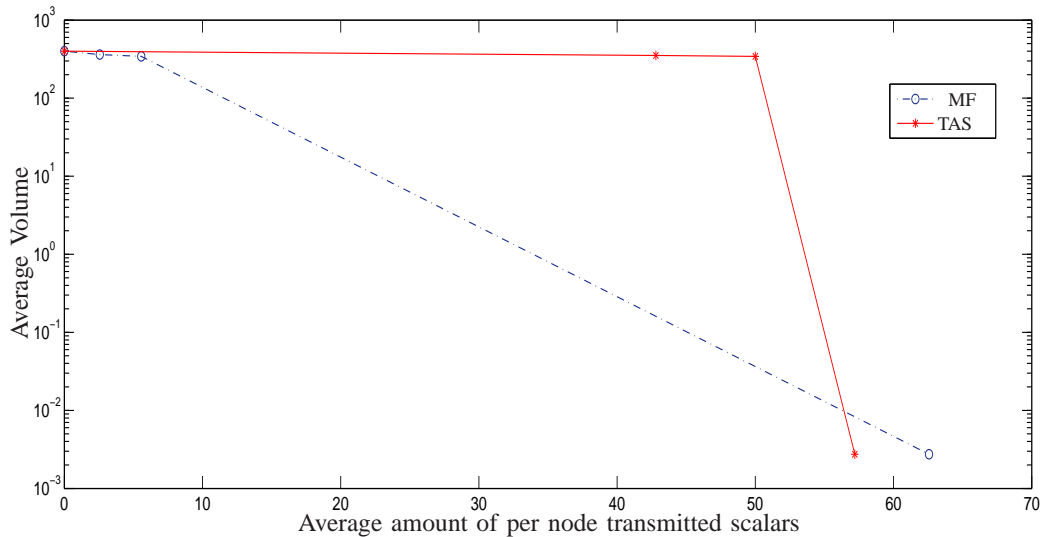


Fig. 8: Average volume, across nodes and 100 clustered network realizations, of the 90% confidence region. Simulation parameters are set to  $n_p = 2$ ,  $q = 1$ ,  $n_c = 20$ , and  $m = 10$ .

## VII. CONCLUSIONS

This paper investigated the distributed evaluation of non-asymptotic confidence regions at each node in wireless sensor networks. The first main contribution is the proposal of the TAS algorithm and its comparison with other information diffusion algorithms on structured and unstructured topologies. The second important contribution consists in demonstrating that, even in presence of truncated information diffusion, the level of confidence remains the same as in the centralized not truncated case. Simulation results provide a characterization of the trade-off for the achievable average confidence region volume as a function of the required amount of data that each node should transmit on average. The contributions nicely concur at showing that, on structured networks, the proposed TAS algorithm is able to outperform the MF when the network dimension is sufficiently high, this independently of the specific estimation problem dimensions, as investigated in the theoretical and numerical analyses. Future research work will be directed at combining the benefits of TAS and MF into a mixed approach, with the scope to further reduce the traffic burden.

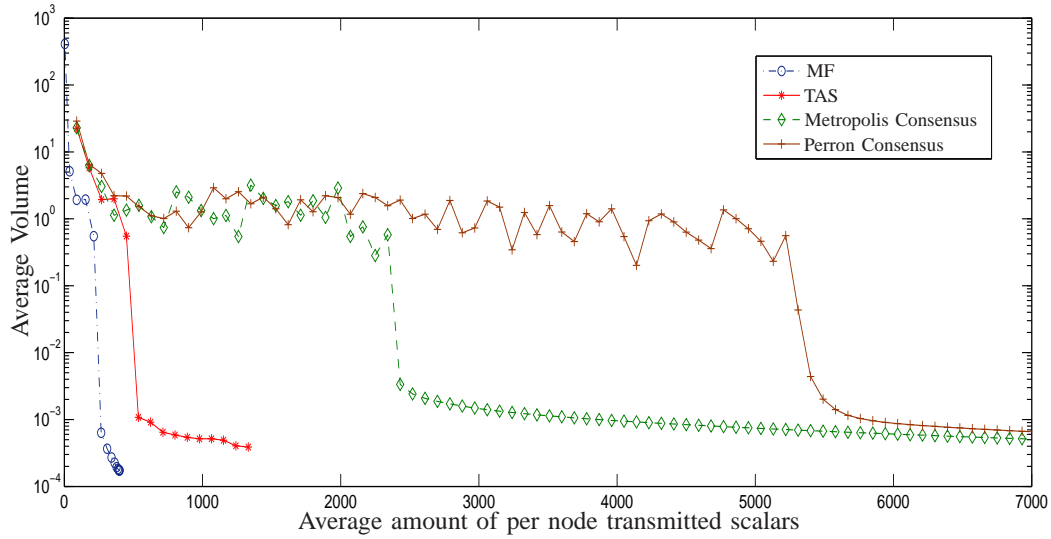


Fig. 9: Average volume, across nodes, of the 90% confidence region. A random unstructured network of 100 nodes is considered.

#### ACKNOWLEDGMENT

This work has been supported by the EU FP7 funded Network of Excellence Newcom#.

#### REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.
- [2] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ser. WSNA ’02. New York, NY, USA: ACM, 2002, pp. 88–97.
- [3] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, in *Wireless Sensor and Actuator Networks: technologies, analysis and design*. Elsevier Ltd, London, 2008.
- [4] T. Quek, D. Dardari, and M. Win, “Energy efficiency of dense wireless sensor networks: to cooperate or not to cooperate,” *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 2, pp. 459–470, February 2007.
- [5] J. Matamoros, F. Fabbri, C. Anton-Haro, and D. Dardari, “On the estimation of randomly sampled 2D spatial fields under bandwidth constraints,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 12, pp. 4184–4192, Dec. 2011.
- [6] G. Mao, B. Fidan, and B. D. Anderson, “Wireless sensor network localization techniques,” *Computer Networks*, vol. 51, no. 10, pp. 2529 – 2553, 2007.
- [7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct 2002.

- [8] R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 8, pp. 2185–2193, Aug 2006.
- [9] Y. Yu, V. Prasanna, and B. Krishnamachari, "Energy minimization for real-time data gathering in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3087–3096, November 2006.
- [10] M. Yu, K. Leung, and A. Malvankar, "A dynamic clustering and energy efficient routing technique for sensor networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 8, pp. 3069–3079, August 2007.
- [11] S. Kwon and N. Shroff, "Energy-efficient unified routing algorithm for multi-hop wireless networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 11, pp. 3890–3899, November 2012.
- [12] S. M. Kay, *Fundamentals of Statistical Signal Processing-Estimation Theory*. Prentice Hall, 2013.
- [13] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, June 2009.
- [14] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," *IEEE Trans. on Signal Processing*, vol. 57, no. 11, pp. 4583–4588, November 2009.
- [15] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.
- [16] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *Decision and Control, 2007 46th IEEE Conference on*, Dec 2007, pp. 5492–5498.
- [17] —, "Kalman-consensus filter: Optimality, stability, and performance," in *IEEE Conf. on Decision and Control Proc.*, Shanghai, China, December 2009, pp. 7036–7042.
- [18] B. Yang and J. Scheuing, "Cramer-Rao bound and optimum sensor array for source localization from time differences of arrival," in *Proceedings of IEEE ICASSP*, March 2005, pp. 961–964.
- [19] X. Sheng and Y.-H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. Signal Processing*, vol. 53, no. 1, pp. 44–53, Jan 2005.
- [20] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 54–69, July 2005.
- [21] D. Jourdan, D. Dardari, and M. Win, "Position error bound for uwb localization in dense cluttered environments," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 44, no. 2, pp. 613–628, April 2008.
- [22] M. C. Campi and E. Weyer, "Guaranteed non-asymptotic confidence regions in system identification," *Automatica*, vol. 41, no. 10, pp. 1751–1764, October 2005.
- [23] M. C. Campi, S. Ko, and E. Weyer, "Non-asymptotic confidence regions for model parameters in the presence of unmodelled dynamics," *Automatica*, vol. 45, no. 10, pp. 2175–2186, October 2009.
- [24] B. C. Csàji, M. C. Campi, and E. Weyer, "Non-asymptotic confidence regions for the least-squares estimate," in *Proc. IFAC SYSID*, Brussels, Belgium, 2012, pp. 227–232.
- [25] M. Kieffer and E. Walter, "Guaranteed characterization of exact non-asymptotic confidence regions as defined by LSCR and SPS," *Automatica*, vol. 50, no. 2, pp. 507–512, February 2014.
- [26] V. Zambianchi, M. Kieffer, F. Bassi, G. Pasolini, and D. Dardari, "Distributed SPS algorithms for non-asymptotic confidence region evaluation," in *Proc. of European Conference on Networking and Communication, EUCNC 2014*, June 2014.
- [27] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ser. MobiCom '99. New York, NY, USA: ACM, 1999, pp. 174–185.

- [28] A. Nordio, C. Chiasserini, and E. Viterbo, "Performance of linear field reconstruction techniques with noise and uncertain sensor locations," *IEEE Trans. Signal Processing*, vol. 56, no. 8, pp. 3535–3547, Aug 2008.
- [29] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb 2003.
- [30] R. Koetter and M. Medard, "An algebraic approach to network coding," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 5, pp. 782–795, Oct 2003.
- [31] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct 2006.
- [32] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, September 2004.
- [33] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. of Information Processing in Sensor Networks, IPSN 2005. Fourth International Symposium on*, April 2005, pp. 63–70.
- [34] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 27–41, 2006.
- [35] L. Xiao, S. Boyd, and S. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributing Computing*, vol. 67, no. 1, pp. 34–46, 2007.
- [36] S. Rump, "INTLAB - INTerval LABoratory," in *Developments in Reliable Computing*, T. Csendes, Ed. Dordrecht: Kluwer Academic Publishers, 1999, pp. 77–104. [Online]. Available: <http://www.ti3.tu-harburg.de/rump/>
- [37] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, Mar 2000.